UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/848,713	05/03/2001	Doug Grumann	10002681-1	7924

22879 7590 10/22/2009

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

EXAMINER	
AILES, BENJAMIN A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
10/22/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
laura.m.clark@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/848,713
Filing Date: May 03, 2001
Appellant(s): GRUMANN ET AL.

John K. Harrop
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 05 August 2009 appealing from the Office action mailed 05 August 2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

.(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,633,908	**LEYMANN**	**10-2003**
5,949,976	**CHAPPELLE**	**9-1999**
6,647,413	**WALRAND**	**11-2003**

Open Group Technical Standard: "Systems Management: Application Response Measurement (ARM) API", July 1998, The Open Group.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1, 2, 4, 5, 6, 8, 11, 12, 14, 17-22 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Systems Management: Application Response Measurement (ARM) API” (The Open Group), hereinafter referred to as “ARM API”, in view of Leymann et al. (US 6,633,908 B1), hereinafter referred to as Leymann.

4. Regarding claim 1, ARM API teaches a method for dynamically determining the health of a service resident on a host machine (page 3, figure 1-1), comprising:

collecting service performance information (p. 3, fig. 1-1, measurement agent) from the resident service (p. 3, fig. 1-1, client, server end systems), wherein the collected service information relates to a plurality of performance metrics (fig. 1-1, monitor application response); and

wherein the output comprises a plurality of service health metrics, and the plurality of performance metrics to provide one or more of the plurality of service health metrics, wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction (fig. 1-1, monitor application response).

ARM API teaches in (fig. 1-1, monitor application response) the collection of service performance information by an ARM API wherein the output is one of a scriptable interface and application programming interface (fig. 1-1, use of ARM API) and useable by different performance monitoring tools (fig. 1-1, measurement agent) but does not explicitly recite the translation of the performance information into a generic output. However, in related art, Leymann teaches these features. Leymann teaches the utilization of an application response measurement API (fig. 1, 106) in communication with an API sub-agent (fig. 1, 107) over a network wherein an agent is utilized for data handling and is deemed generic in order to be independent from a specific application and therefore the data is available for all applications requesting the data (col. 8, ll. 3-14). In view of Leymann, it is therefore deemed that it would have been

obvious to one of ordinary skill in the art to implement the ARM API to translate the collected service performance information in to a generic output. One of ordinary skill in the art would have been motivated to incorporate the teachings of Leymann with ARM API in order to implement independence from a specific application and make the data available for a wide range of calling applications (Leymann, col. 8, ll. 9-13).

5. Regarding claim 2, ARM API and Leymann teach the method wherein the host machine comprises one or more components, further comprising:

collecting external performance information from one or more of the one or more components (ARM API, fig. 1-1, monitor application response);

translating the collected external performance information (Leymann, col. 8, ll. 9-13); and

combining the translated external performance information and the translated service performance information to provide the generic output (Leymann, col. 8, ll. 9-13).

6. Regarding claim 4, ARM API and Leymann teach the method further comprising accessing the generic output to read the health of the service (Leymann, col. 8, ll. 9-13).

7. Regarding claim 5, ARM API and Leymann teach the method wherein the collecting step comprises reading performance information provided by the service (ARM API, fig. 1-1, monitor application response).

8. Regarding claim 6, ARM API and Leymann teach the method wherein the collecting step comprises deriving performance information from the service (ARM API, fig. 1-1, monitor application response).

9. Regarding claim 8, ARM API and Leymann teach the method wherein the deriving step comprises using a probe program to read the performance information (Leymann, col. 8, ll. 9-13, data read by independent components).

10. Regarding claim 11, ARM API teaches an apparatus that determines a health of a service resident on a host machine, comprising:

a data collection engine (p. 3, fig. 1-1, measurement agent) that collects service health information (fig. 1-1, monitor application response);

wherein the collected service health information relates to a plurality of performance metrics, the plurality of performance metrics to provide one or more of the plurality of service health metrics, wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction (fig. 1-1, monitor application response).

ARM API teaches in (fig. 1-1, monitor application response) the collection of service performance information by an ARM API wherein the output is one of a scriptable interface and application programming interface (fig. 1-1, use of ARM API) and useable by different performance monitoring tools (fig. 1-1, measurement agent) but does not explicitly recite the translation of the data in a health generation algorithm providing one or more generic health metrics. However, in related art, Leymann teaches these features. Leymann teaches the utilization of an application response measurement API (fig. 1, 106) in communication with an API sub-agent (fig. 1, 107) over a network wherein an agent is utilized for data handling and is deemed generic in order to be independent from a specific application and therefore the data is available

for all applications requesting the data (col. 8, ll. 3-14). In view of Leymann, it is therefore deemed that it would have been obvious to one of ordinary skill in the art to implement the ARM API to translate the collected service performance information in to a generic output. One of ordinary skill in the art would have been motivated to incorporate the teachings of Leymann with ARM API in order to implement independence from a specific application and make the data available for a wide range of calling applications (Leymann, col. 8, ll. 9-13).

11. Regarding claim 12, ARM API and Leymann teach the apparatus wherein the host machine comprises one or more external components, wherein the data collection engine collects external performance information from one or more external components (ARM API, fig. 1-1, monitor application response) and wherein the data analysis engine translates the collected external information using the health generation algorithm to provide the one or more generic health metrics (Leymann, col. 8, ll. 3-14).

12. Regarding claim 14, ARM API and Leymann teach the apparatus wherein the data collection engine, comprises:

a data query module that reads performance information from the service (ARM API, fig. 1-1, measurement agent); and

a data derivation module that derives performance information from the service (ARM API, fig. 1-1, monitor application response).

13. Regarding claim 17, ARM API and Leymann teach the apparatus further comprising an interval control engine that receives the service health information at a

first time interval and provides an output having a second time interval different from the first time interval (Leymann, col. 8, ll. 3-7).

14. Regarding claim 18, ARM API teaches an apparatus that determines a health of a service resident on a host machine, comprising:

a data collection engine (p. 3, fig. 1-1, measurement agent) that collects service health information (fig. 1-1, monitor application response);

wherein the collected service health information relates to a plurality of performance metrics, the plurality of performance metrics to provide one or more of the plurality of service health metrics, wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction (fig. 1-1, monitor application response).

ARM API teaches in (fig. 1-1, monitor application response) the collection of service performance information by an ARM API wherein the output is one of a scriptable interface and application programming interface (fig. 1-1, use of ARM API) and useable by different performance monitoring tools (fig. 1-1, measurement agent) but does not explicitly recite the translation of the data in a health generation algorithm providing one or more generic health metrics. However, in related art, Leymann teaches these features. Leymann teaches the utilization of an application response measurement API (fig. 1, 106) in communication with an API sub-agent (fig. 1, 107) over a network wherein an agent is utilized for data handling and is deemed generic in order to be independent from a specific application and therefore the data is available for all applications requesting the data (col. 8, ll. 3-14). In view of Leymann, it is

therefore deemed that it would have been obvious to one of ordinary skill in the art to implement the ARM API to translate the collected service performance information in to a generic output. One of ordinary skill in the art would have been motivated to incorporate the teachings of Leymann with ARM API in order to implement independence from a specific application and make the data available for a wide range of calling applications (Leymann, col. 8, ll. 9-13).

15. Regarding claim 19, ARM API and Leymann teach the method wherein the step of collecting the service performance information comprises reading first service performance parameters, and wherein the step of collecting the external performance information comprises reading first external performance parameters and deriving second external performance parameters (ARM API, fig. 1-1, monitor application response).

16. Regarding claim 20, ARM API and Leymann teach the method further comprising collecting the service performance information on a first time interval and adjusting the first time interval to provide the generic service health output at a second time interval (Leymann, col. 8, ll. 3-7).

17. Regarding claim 21, ARM API teaches an apparatus that determines a health of a service, wherein the service operates on a host computer (page 3, figure 1-1), comprising:

a collection module that receives performance information related to the service (p. 3, fig. 1-1, measurement agent); and

wherein the output comprises a plurality of service health metrics, and the plurality of performance metrics to provide one or more of the plurality of service health metrics, wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction (fig. 1-1, monitor application response).

ARM API teaches in (fig. 1-1, monitor application response) the collection of service performance information by an ARM API wherein the output is one of a scriptable interface and application programming interface (fig. 1-1, use of ARM API) and useable by different performance monitoring tools (fig. 1-1, measurement agent) but does not explicitly recite the translation of the performance information into a generic output. However, in related art, Leymann teaches these features. Leymann teaches the utilization of an application response measurement API (fig. 1, 106) in communication with an API sub-agent (fig. 1, 107) over a network wherein an agent is utilized for data handling and is deemed generic in order to be independent from a specific application and therefore the data is available for all applications requesting the data (col. 8, ll. 3-14). In view of Leymann, it is therefore deemed that it would have been obvious to one of ordinary skill in the art to implement the ARM API to translate the collected service performance information in to a generic output. One of ordinary skill in the art would have been motivated to incorporate the teachings of Leymann with ARM API in order to implement independence from a specific application and make the data available for a wide range of calling applications (Leymann, col. 8, ll. 9-13).

18. Regarding claim 22, ARM API and Leymann teach the apparatus wherein the collection module receives external performance information from one or more external services coupled to the host computer and receives internal performance information related to operation of the service on the host computer (ARM API, fig. 1-1, monitor application response).

19. Regarding claim 24, ARM API and Leymann teach the apparatus wherein the generic health metrics is one of a scriptable interface and an application programming interface (ARM API, use of API for response measurement).

20. Claims 7 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over ARM API and Leymann in view of Chappelle (US 5,949,976).

21. Regarding claim 7, ARM API and Leymann do not explicitly teach of using a wrapper program. Chappelle teaches about using a wrapper program (performance monitoring and graphing tool) to read the performance information (col. 3, ll. 29-32). The examiner is interpreting wrapper program as any program that is used as an interface program because this gives the broadest reasonable interpretation. In ARM API's specification, the performance forecasting system communicates with one or more monitoring system (fig. 1-1, enterprise management solutions). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to utilize the teaching of Chappelle in regards to using a wrapper program because it would have allowed the performance forecasting system to read the information supplied by various monitoring systems regardless of the components particular

infrastructure. One of ordinary skill in the art would have been motivated because this modification would result in a more versatile system as outlined above.

22. Regarding claim 15, ARM API and Leymann do not explicitly teach of using a wrapper program. Chappelle teaches about using a wrapper program (performance monitoring and graphing tool) to read the performance information (col. 3, ll. 29-32). The examiner is interpreting wrapper program as any program that is used as an interface program because this gives the broadest reasonable interpretation. In ARM API's specification, the performance forecasting system communicates with one or more monitoring system (fig. 1-1, enterprise management solutions). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to utilize the teaching of Chappelle in regards to using a wrapper program because it would have allowed the performance forecasting system to read the information supplied by various monitoring systems regardless of the components particular infrastructure. One of ordinary skill in the art would have been motivated because this modification would result in a more versatile system as outlined above.

23. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over ARM API and Leymann in view of Walrand et al. (US 6,647,413), hereinafter referred to as Warland.

24. Regarding claim 16, ARM API and Leymann do not explicitly teach of a weighting scheme that weights one or more performance information parameters; a summation scheme that combines one or more performance information parameters; and a averaging scheme that averages collected service health information for a service

health metric. However, Walrand teaches on these aspects. Walrand teaches about a summation scheme that combines one or more performance information parameters (col. 7, ll. 32-33) and an averaging scheme that averages collected service health information for a service health metric (col. 7, ll. 55-57). In HPCN Walrand teaches of a weighting scheme that allocates different level of importance to different parameters (p. 2). One objective of Walrand invention is to optimize the network performance (col. 2, ll. 53-54). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to utilize the above mentioned features of Walrand's into ARM API and Leymann because adding these features to ARM API and Leymann would allow focus on specific parameters (using the weighting scheme) and give information regarding the overall performance of the network system (using the summation and averaging schemes). These added features would allow ARM API and Leymann to provide a healthy network and more effectively predict failure of registered computing devices (col. 2, ll. 25-34) resulting in a more efficient performance forecasting system. It is for this reason that one of ordinary skill in the art at the time of invention would have been motivated to make the above-mentioned modifications.

(10) Response to Argument

A. Rejection of claims 1, 2, 4-6, 8, 11, 12, 14, 17-22 and 24 under 35 USC 103(a)

Claims 1, 11, 18 and 21

With respect to the rejection of claim 1 under 35 USC 103(a) in view of "Systems Management: Application Response Measurement (ARM) API" (The Open Group), hereinafter referred to as "ARM API", in view of Leymann et al. (US 6,633,908 B1), the

appellant argues (a) that the cited prior art does not disclose or suggest the use of "a generic output" as claimed and (b) that Leymann and ARM API cannot be used in combination.

(a) In response to argument (a), the examiner respectfully disagrees and submits that the prior art as applied in the rejections teaches within the claimed scope sought by appellant. With respect to teaching the "use of a generic output" the examiner maintains that Leymann teaches on this broadly claimed aspect wherein Leymann teaches in column 8, lines 3-14 the making of data available for all applications requesting the data by use of an invocation agent. The invocation agent provides the data needed by calling applications and is considered a generic component due to its independence from specific applications. Because of the independence of the invocation agent, the data it provides is therefore considered to be "generic output" as required by the claims. The output provided by the invocation agent is deemed generic because it is made available for all applications. The ARM API non-patent reference is relied upon for teaching the further aspect of having the output data in a scriptable interface for application programming interface wherein ARM API teaches in Fig. 1-1 on page 3 the use of the application response measurement API. The claim provides no further guidance as to how a generic output is utilized beyond the generic output "relating to current operational performance of the service resident on a host machine, the generic output comprising a plurality of service health metrics." The examiner submits that the generic output representing a plurality of service health metrics with respect to current operational performance parameters is taught by at least ARM API on page 3, figure 1-1

by the collection of application response parameters which is within the claimed one or more plurality of service health metrics (availability, capacity, throughput, service time, queue length, utilization, service level violations and user satisfaction.).

(b) In response to appellant's argument (b) that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the examiner maintains that one of ordinary skill would have been motivated to make a data output a generic form as taught by Leymann. The examiner maintains that one of ordinary skill would have been motivated to utilize Leymann in combination with the ARM API reference for the same reasons set forth in argument (a) and specifically implement independence from a specific application and make the data available for a wide range of calling applications as taught by Leymann in column 8, lines 9-13.

Therefore, the examiner submits that the combination of ARM API and Leymann is found to teach appellant's claimed usage of "generic output" and the rejection should be sustained. Appellant's arguments for independent claims 11, 18 and 21 are similar in scope to claim 1 and therefore the rejections should be sustained for the same reason(s).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Benjamin A. Ailes
/BAA/
Examiner, Art Unit 2442

/Jeffrey Pwu/
Supervisory Patent Examiner, Art Unit 2446

Conferees:

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2446

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451